APPENDIX B

FUNCTIONAL UNITS

APPENDIX B

FUNCTIONAL UNITS

Each RDA functional unit is described below in alphabetical order. Appendix D, RDA Objects Calling Other Objects, contains the mappings of each application to other applications, database tables, and database packages that it calls. Appendix E, RDAObjects Called By Other Objects, lists each of these applications in relationship to the other modules (packages or Gain applications) that call it.

B.1 ArrayManager

This Gain tool displays a list of various information on all the requirements in the current collection. It can also list the contents of any other specified array.

B.2 BrowserTool

This is a Gain helper tool. This tool allows Gain Developers to browse through various objects. Double clicking on an object in the object list retrieves the children of the selected object. In this manner the developer can traverse through any object hierarchy, to search for a particular object, and then view the properties of a selected object. Some of the functionality which exists in this tool includes:

a. Expand children of an object;

b. Expand ALL children of an object (allowing the developer to view all objects contained in a page or application);

c. Print the scripts of listed objects;

d. Print the scripts of listed objects and ALL children;

e. View an object's property list;

f. View specific properties of an object; and

g. Edit / Modify an object property.

B.3 ButtonBar

This is a Gain helper tool, which simply allows the developers quick access to the various development tools built in the Gain development environment. The tool contains buttons for several development tools. When the developer wishes to start one of these tools, the buttonPress action causes the "ButtonBar" to send the "openTool" message or popupPage() function call to the selected tool to bring the tool up on the developer's screen. Similarly, a second buttonPress action will trigger the "closeTool" message or "popdownPage()" function to be called to the selected tool to close it and remove it from the developer's screen.

B.4 CargoEditor

The Cargo Editor is run from within the Edit TPFDD function. This application allows the RDA user to VIEW and EDIT ULN Cargo Detail information for the selected OPLAN. The Cargo Editor operates on the requirements placed in the RDA_COLLECTION table by RDA's Select Functional Unit. Force Requirement detail cargo information can be viewed and edited at the third and fourth levels.

To enter the Cargo Editor Functional Unit, the user selects Choose Display >> Cargo Editor from the main menu bar in the Edit TPFDD screen. The Force Requirement ID's and ORACLE ROWIDs for all ULNs (except those with Force Indicator Code (FIC) of 7) in the collection table are fetched by the Cargo Editor. Cargo Editor uses this information to retrieve detail cargo information and display it to the user.

B.5 CargoSelect

This functional unit provides a list of the three cargo category codes and their meanings. In addition, the valid combinations of the three are listed and the user can select a combination to use in creation of a cargo record.

B.6 CINDetails

This function unit provides a method for the user to display and edit detail information for an individual CIN requirement. After data is added or changed, applying the changes creates the appropriate transaction. Applicable force modules can also be listed.

B.7 CollectionDetail

This permits a user to apply global edits of requirement fields to all requirements (either all ULNs, CINs, or PINs) in a collection.

B.8 Colors

This application displays the current and old color schemes for RDA.

B.9 ComparePIDs

The Compare TPFDDs functional unit is an application designed to Compare a source OPLAN to a target OPLAN. The comparison of data can be on all Services and Providing Organizations or a subset thereof. Likewise, a subset of fields may be chosen in the "Selective Compare" mode, or all pre-determined fields may be compared by choosing the "Full Compare" option.

In any event, the Compare TPFDDs FU is designed to compare designated fields between the Source and Target TPFDDs, and if there any differences (case-sensitive), they will be eventually displayed in the F52 pre-defined report. Also, the Compare TPFDDs FU will identify Location records (e.g., Location records in the OPLAN_FORCE_RQMT_LOC table) that are NOT found in one OPLAN and found in the other, if fields being compared are Not NULL or non-blank in the existent record. If a field is NULL or blank and the corresponding field to compare is NOT found, due to a non-existent Location record, they will NOT be considered different. Therefore, the information will not be recorded, and subsequently displayed in the pre-defined report generated (F52).

The Compare application is called from the RDA Skeleton main window by selecting the "Compare Plans"

option from the "Selected" main menu selection. Once the Compare application is popped up on the screen, the source PID to Compare will automatically be the PID selected from the RDA Skeleton main window.

The Compare function has three phases. The first phase pulls out the requirements that exist in both the source and the target plans. The second phase uses these requirements to actually retrieve data to compare between the two plans. Both phases run in the background. When done, they generate pop-up windows (if the GAIN session is still active) and send e-mail to the user. The third phase invokes SQLReportWriter to print the report on screen or on paper.

B.10 CopyPlan

The CopyPlan function allows a user to copy data information from a source OPLAN into an existing target OPLAN. The target OPLAN need not be empty. The process includes 3 phases, invoked from GEL code:

Phases:

a. The transaction recording into the "send_queue" table of data to be deleted from the target OPLAN and data to be inserted from the source OPLAN to the target OPLAN.

b. The deletion of existing data from the target OPLAN.

c. The insertion of the source OPLAN data into the target OPLAN.

The process does not include S&M data. It includes the OPLAN Narrative, Summary, Requirements, and Force Modules. It runs in the background after the user selects the target plan. When done, it creates a pop-up window (if the GAIN session is still active) and sends e-mail to the user. If a plan is used as a target plan, that plan cannot be used in another CopyPlan process until the existing process is finished. In the case where the plan id is blocked even after the process is done, from SQLPLUS, delete from the table rda_parameter_list where rda_arg3 = 'Z' and user_id = theuser.

B.11 CreateCINsPINs

This permits the user to create up to 99 CINs or PINs at a time. The user chooses a Using Organization and Type of Movement which are both characters representing the first two characters of a CIN or PIN. Cargo category codes must be chosen for the CINs. Requirements are numbered sequentially from a seed value which must 7 characters in length. By default, the new requirements are added to the users collection.

B.12 CreateRecs

This allows the user to create ULNs, CINs, or PINs.

B.13 CreateULNs

This permits the user to create up to 99 ULNs at a time. The user chooses a UTC value that all the ULNs created will contain. Requirements are numbered sequentially from a seed value which can be up to 7 characters in length. By default, the new requirements are added to the users collection.

B.14 EditForceModule

This functional unit provides Force Module (FM) editing and optional FM operations. Force module editing provides the following operations: creation of a force module, addition of requirements to a force module, removal of requirements from a force module, replacement of the contents of a force module, modification of a force module's title, importing, modification, and exporting of a force module's description, removal of a force module from a PID, duplication of the requirements in one or more force modules, copy of force modules in one PID to another PID, creation of a new force module with the same title and description as an existing force module, renaming of a force module, renumbering of the requirements in one or more force modules, marking of all requirements in the current collection that are in a force module, and FM reports. In order to produce a FM report, EditForceModule accesses functional unit Marked_Records of work package RDARequirements. Marked_Records then accesses the Pre-Defined Reports (PDR) segment.

Optional FM Operations are available to the user after a successful completion of the following RDA operations: creation of requirements, renumbering of requirements, duplication of requirements, or copy of marked requirements to target PID with option "Optional FM Operations" selected. The optional FM operations provided are the following: creation of a new force module containing the requirements, addition of the requirements to an existing force module, or replacement of the contents of an existing force module with the requirements. The functional units that access EditForceModule for optional FM operations are RenumberOps, Marked_Records, CreateCINsPINs, and CreateULNs.

B.15 Editor

The Editor functional unit is the main application that allows the user access to functions that modify/create requirements and force modules. The user can call the Select function to define a collection of records to work with. Collections are displayed by either calling the Timeline or Cargo Editor functions. Operations called involving requirements include creating, template editing, individual editing, copying to a target OPLAN, validating the data, shifting dates, and linking to force modules. Force module editing and graphical display functional units can also be called. In addition, hot key rapid navigation capability is available for detail editing, renumbering, duplicating, and deleting operations on requirements.

B.16 EditRecords

This application permits the user to modify, delete, or duplicate requirements (ULNs, CINs, or PINs) from user's collection of requirements. Selections are made by clicking the right mouse button on the requirement in the Timeline or clicking the right mouse button on a ULN in the Cargo Editor.

The following are the operations allowed on selected requirements in the user's collection:

Key: U = ULNs
C = CINs
P = PINs

	Operation	**Requirement Values Operated On**
a.	Unmark	(U, C, P)

Unmarks selected requirement currently in the collection.

b. Mark All (U, C, P)
Marks selected requirement currently in the collection.

c. Remove (U, C, P)
Removes selected requirement currently in the collection.

d. Details... (U, C, P)
Allows the user to make edits on contents of requirement.

e. Delete (U, C, P)
Deletes selected requirement from the OPLAN.

f. Duplicate... (U, C, P)
Allows the user to make copies of selected requirement.
Duplicated records are added to the same FM(s) as the original record.
New records are added to the collection.

g. Renumber... (U, C, P)
Allows the user to renumber selected requirement in the collection.

h. Fragment/Insert... (U)
Allows the user to generate fragment and insert records from the selected requirement.

I. Split Shipments... (U)
Allows the user to split selected ULN.

j. Unsplit Shipments... (U)
Allows the user to unsplit selected ULN.

B.17 Engine Helper

This tool is provided to aid in the maintenance of the RDASelect work package.

B.18 EquipmentLookup

This application provides a lookup for equipment information (TUDET) for the user to select when creating a cargo detail record.

B.19 ErrorSelect

This function is used to allow the user to browse the list of logical error codes and pick one or more error codes to be used in a query.

B.20 FileSelector

This application is a system tool, which is accessed and used by various other Gain tools to access files in the UNIX file system. The purpose of this application is to provide the developers a Graphical User Interface to traverse the directory structure, to aid in locating or creating a valid file reference (i.e., file name and path) and return to the calling function.

B.21 Finder

This tool will search through any specified Gain hierarchy and find all occurrences of a specified string.

B.22 FMdata

This function provides functions for retrieving and manipulating force module data.

B.23 FMGraphics

This function provides the main display portion of the user interface for Flow Analysis. This display is a grid with deployment days across the top and force module information down the side.

B.24 FMlist

This FU provides the functions and user interface for selecting the desired force modules for display.

B.25 FMwindows

This FU provides supporting functions and windows for flow analysis.

B.26 GainMetrics

This tool gathers metrics (counts for windows, menus, datamanagers, and GEL code) for any selected application. Metrics can be stored in a buffer and exported to a comma-delimited file for further manipulation in a spreadsheet.

B.27 GEOSelect

This application provides the user a search and lookup capability for geographic location (GEOLOC) information for requirement routing data.

B.28 Grep

This allows the user to search for string patterns in Gain Momentum script files. From a message box the usage is: app grep of library RDADevTools.grep (<obj_name>, <string>, <grep_flags>, <traverse>), where <obj_name> is the object reference of a Gain Momentum object, <string> is the string searched for, <grep_flags> is a string representing any flags the UNIX grep utility takes as arguments (precede flags with -), and <traverse> is a boolean indicating if the objects' children are to be traversed.

B.29 GSORTSSelect

This application provides the user a search and lookup capability for unit data by referencing the GCCS Status of Resources and Training System (GSORTS) for ULN sourcing.

B.30 Heap

This application displays the amount of memory that is being used by the Gain applications.

B.31 LookUpApp

This application accepts column and table parameters and uses them to query the specified table and display the column values and their meanings to the user for selection.

B.32 Marked_Records

This permits the user to do the following operations on marked requirements in the collections:

Key: U = ULNs
C = CINs
P = PINs

	Operation	**Requirement Values Operated On**
a.	Unmark All	(U, C, P)
	Unmarks all requirements currently in the collection.	
b.	Mark All	(U, C, P)
	Marks all requirements currently in the collection	
c.	Toggle All	(U, C, P)
	Toggles the marked state of all requirements in the collection.	
d.	Marked Inserted	(U, C, P)
	TBD	
e.	List...	(U, C, P)
	Lists all marked requirements currently in the collection.	

f. Delete (U, C, P)
Deletes all marked requirements currently in the collection.

g. Remove (U, C, P)
Removes all marked requirements currently in the collection.

h. Duplicate... (U, C, P)
Allows the user to make copies of all marked requirements in the collection. Duplicated records are added to the same FM(s) as the original record.

i. Renumber... (U, C, P)
Allows the user to renumber marked requirements in the collection.

j. ULN Details... (U)
Allows the user to make range updates to requirement and routing data for all marked ULNs in the collection.

k. CIN Details... (C)
Allows the user to make range updates to requirement and routing data for all marked CINs in the collection.

l. PIN Details... (P)
Allows the user to make range updates to requirement and routing data for all marked PINs in the collection.

m. Split Shipments... (U)
Allows the user to split marked ULNs in the collection.

n. Unsplit Shipments... (U)
Allows the user to unsplit marked ULNs in the collection.

o. Reports... (U, C, P)
Allows the user to generate reports on marked requirements in the collection.

p. Copy To Target... (U, C, P)
Allows the user to copy marked requirements to another PID.

B.33 MergePIDs

The Merge TPFDDs functional unit is an application designed to merge a source OPLAN to a target OPLAN. The movement of data is in one direction (i.e., from the source OPLAN to the target OPLAN). This means that any requirements in the target OPLAN not found in the source OPLAN (before the merge) will remain in the target OPLAN. Likewise, any requirements found in the source OPLAN and not found in the target OPLAN, will be added to the target OPLAN after the merge operation. Therefore, it is important to note that the Merge process is NOT simply a Copy TPFDD from source OPLAN to target OPLAN.

The Merge process has an automatic or manual mode for handling duplicate requirements data. If there are requirements to be merged that are common to both Source and Target OPLANs, then the user may either (1) Replace the target requirement with the source requirement, (2) Skip the target requirement (i.e., do not overwrite the target requirement with the source requirement), and (3) Renumber any of these common requirements before executing the merge process.

In the manual merge option, the user will first be prompted to manally merge requirements common to both Source and Target OPLANs. In this case, the user will be asked to either (1) Replace target requirement with the source requirement, (2) Keep the Target requirement (i.e., do NOT overwrite target requirement with source requirement), and (3) Enter a unique requirement to use as the new target requirement. After the user is finished with the manual process, all requirements that are NOT common to the Source and Target OPLANs are automatically merged en masse.

The Merge application is called from the RDASkeleton main window by selecting the "Merge Plans" option from the "Selected" main menu selection. Once the Merge application is popped up on the screen, the source PID to merge will automatically be the PID selected from the RDA Skeleton main window.

The Automatic Merge function executes in the background. When done, it creates a pop-up window (if the GAIN session is still active) and send e-mail to the user. If a plan is used as a target plan, it cannot be used in another MergePlan process until the existing process is finished. In the case where the plan id is blocked even after the process is done, from SQL PLUS, delete from the table rda_parameter_list where rda_arg3 = 'W' and user_id = theuser.

B.34 MiscOps

This tool is a modified version of the tool described in the Gain Manual "Tool Builders Guide." Functions are provided for exporting applications/libraries, globally changing permissions, and performing mass check-ins.

B.35 ObjectSelect

This tool provides a hierarchical method of selecting a Gain object..

B.36 OPS$Tool

The environment variables necessary for supporting ORACLE OPS$ accounts can become quite complicated and extensive in size. This tool allows a user to easily view the environment variable definitions in a C-Shell Script file and to modify those variable definitions.

B.37 PagePrivTool

This tool supports the modification of windows to prevent users without update permission from accessing functions that update the database. Basically, this tool can be used to generate a property on a page that will be used by the SetPagePermissions function within RDAToolBox to modify a window to prevent database updates from occurring (i.e. making required fields read-only, and deactivating buttons).

B.38 PINDetails

PINDetails provides a method for the user to display and edit detail information for an individual PIN requirement. Data to be edited is pulled from OPLAN_NONUNIT_RQMT_LOC and OPLAN_NONUNT_RQMT_PRSL. Edit checks are applied on all fields for data type, and where applicable, code. After data is added or changed, applying the changes creates the appropriate transaction. Applicable FMs can also be listed.

B.39 PlanSummary

The PlanSummary functional unit allows viewing and selected editing of OPLAN/TPFDD related information. Updateable information on the first page includes title, classification (if user is originator of the plan), objective area, mobilization lead time and mobilization required. When any of these fields are updated, the changed date fields is also modified with the current date/time. Upon pressing apply, database update and transaction generation occurs. The Plan Information (PLNUAT) transaction is generated through a call to the RDA_TRANSACTION function in RDAToolBox. The data changed will determine the type of PLNUAT format that is created. Other non-updateable data includes plan status, distribution, number of ULNs, CINs, PINs, Cargo Category Codes (CCCs), FMs, cargo/PAX, GEOLOC counts, and TUCHA date information. Other pages allow the user to view expanded cargo/PAX information for modes of air and sea. In addition, two pop-up pages allow view and update of the Supporting Commanders and Supporting Plans information. The rest of the narrative data (e.g., Mission, Narrative, Concept of Operations, etc.) is viewed and updated from a single reusable screen that also allows importing and exporting of ASCII text from and to UNIX system files. PLNUAT transactions are also generated for the other updateable information and the change date on the first page is also modified.

B.40 PrintMonitor

This tool is used for monitoring the print queus.

B.41 ProgressBar

This application contains the progress bar, which is displayed by various RDA applications to show the users progress being made at various time consuming tasks (i.e., for loops or other iterative processes.) It is accessed by popping up its main page, and updated by sending a function call "UpdateStatus()" to the application script, passing in the total number of iterations to be executed and the current iteration.

B.42 ProfileTool

This tool provides information about timing of Gain functions and is used to analyze performance.

B.43 PropertyManager

This tool allows the developer to monitor property values.

B.44 Pscr

This allows the user to print scripts of Gain Momentem objects.

B.45 RDACollectionCheck

This function is a monitoring function for looking at the records that are in the RDA_COLLECTION table. Periodically, this table accumulated orphaned records, and this RU allows a developer to look at the collection table and remove orphaned records.

B.46 RDAError

This function provieds a standard way of displaying internally (Gain) generated errors to the user and logging those errors to the RDA error log file located in the user's home directory.

B.47 RDASkeleton

RDASkeleton is the main application for entering RDA. Upon initiation of the application the user's permissions are check for access and if the checked is passed a list of all OPLANs that the user has access to are presented. The user may perform wildcard queries on the OPLANs in the database to filter the list.

From the list the user may choose an OPLAN that he wants to work with. There are various operations that may be performed on the OPLAN by selecting the appropriate menu option:

a.	Summary -	View/Update the OPLAN summary/TPFDD information;
b.	Copy Plan-	Copy the contents of the OPLAN to another OPLAN;
c.	Edit TPFDD -	Edit the TPFFD (requirements and force modules);
d.	Compare Plans -	Compare two OPLANs;
e.	Merge Plans -	Merge the contents of two OPLANs; and
f.	Update PID from TUCHA -	Update TUCHA dependent information in the OPLAN.

These operations call other RDA applications. The user may also choose to view the available reference files (e.g., GSORTS, GEOLOC, TUCHA, etc). The screen may be minimized but it remains active for the duration of the user's RDA session. If it is closed then all other RDA windows are also closed after checking if any updates are pending on the windows. If there are updates pending, then the user is prompted to apply changes, cancel changes or return to the RDA execution (cancel the exit operation). User permission changes are also accessed through this application for the entire RDA System. If the user loses or gains update privileges, then the icon at the bottom right of the screen will reset the access to updateable fields and certain menu options globally throughout RDA.

B.48 RDATimeline

RDATimeline's main purpose is to display Requirements (ULNs, CINs, and PINs) in the timeline graphical format.

The user calls RDATimeline from application "Editor." He then presses the "Select Records" pulldown menu to choose a collection of requirements. These requirements are displayed in the timeline graphical format. The user can scroll horizontally, vertically, or zoom to see the requirements he is most interested in displaying. He can click on individual icons to get a popped up window displaying additional details on the icon. In this window the user can make changes to the icon. The minitimeline can be called by the user by clicking on an empty space on the timeline grid. Error flags are displayed to alert the user of unusual conditions in each requirements. Other symbols are also displayed to indicate shortfall, parent, on-call, in-place and airdrop conditions.

For the design of Timeline, Application "Editor" calls application "TLUlnList." Application "TLUlnList" then acts like the "main" program and calls functions in application RDATimeline to populate an array and position graphics. This data array is a global variable and is used throughout the applications in the RDATimeline libraries. Popped up windows displaying additional details on the icons are in the application "TimelineWindows." The minitimeline window is also in "TimelineWindows".

Throughout the program the following global data array is used:

Array row(index)	ORACLE Tables RQMT=OPLAN_FORCE_RQMT table(ULNs) RQMT=OPLAN_NONUNIT_RQMT_PRSL(PINs) RQMT=OPLAN_NONUNIT_RQMT_CARGO(CINs) LOC=OPLAN_FORCE_RQMT_LOC table(ULNs) LOC=OPLAN_NONUNIT_RQMT_LOC(CINs+PINs) COLLECTION=RDA_COLLECTION table

Index Name	**Table Name\ Field Name**	**Format**
ULN_name	LOC\OP_MVTRQT_ID	V 7
rqmt_type.(U,P,C)	COLLECTION\OP_MVTRQT_TY_CD	V 1
ORIG_name	LOC\OP_RTG_GLC_CD	V 4
ORIG_cday	RQMT\OPFRQ_UNIT_RLD_CQY(ULN only)	N 3
POE_name	LOC\OP_RTG_GLC_CD	V 4
POE_cday	RQMT\OP_MVTRQT_POE_ALD_CQY	N 3
to_POE_color	LOC\OP_RTG_TRNPN_MD_CD	V 1
ILOC_pos	LOC\OP_RTG_TRNPN_LOC_CD	V 1
ILOC_name	LOC\OP_RTG_GLC_CD	V 4
ILOC_width	LOC\OP_RTG_FORCE_DELY_DAY(ULNs)	N 3
	LOC\OP_RTG_NONUNIT_DELY_DAY_CQY(C+P)	
to_ILOC_color	LOC\OP_RTG_TRNPN_MD_CD	V 1
POD_name	LOC\OP_RTG_GLC_CD	V 4
POD_ead	RQMT\OP_MVTRQT_POD_EAD_CQY	N 3
POD_lad	RQMT\OP_MVTRQT_POD_LAD_CQY	N 3
to_POD_color	LOC\OP_RTG_TRNPN_MD_CD	V 1
DEST_name	LOC\OP_RTG_GLC_CD	V 4
DEST_cday	RQMT\OP_MVTRQT_DEST_RDD_CQY	N 3
to_DEST_color	LOC\OP_RTG_TRNPN_MD_CD	V 1
CRD_cday	RQMT\OPFRQ_CRD_CQY(ULNs only)	N 3
redFlag	RQMT\OP_MVTRQT_LOG_ERR_CD	V 1
yellowFlag	RQMT\OP_MVTRQT_LOG_ERR_CD	V 1

shortfall	RQMT\OPFRQ_PRVDNG_ORG_SRC_CD(ULN)	V 1
	RQMT\OPNRQ_PRSL_PRV_ORG_SRC_CD(P+C)	V 1
onCall	RQMT\OP_MVTRQT_POD_LAD_CQY	N 3
	RQMT\OP_MVTRQT_DEST_RDD_CQY	N 3
parent	RQMT\OPFRQ_PRNT_CD(ULNs only)	V 1
inPlace	LOC\OP_RTG_TRNPN_MD_CD	V 1
airDrop	LOC\OP_RTG_LD_CFGN_CD (ULN only)	V 1
marked	COLLECTION\OP_MVTRQT_MARK_ICD	V 1
wasDeleted	RQMT\OP_MVTRQT_ID	V 7
validStatus	RQMT\OPFRQ_VLD_STAT_CD(ULNs only)	V 1
validError	RQMT\OPFRQ_VLD_ERR_CD	V 2

For example, the 5th requirement's name in the current buffer is:

```
global RDA_TimeLine_Array          -- This must be the first line in
                                   -- any function or handler that uses
                                   -- the RDA_TimeLine_Array global
                                   -- variable.
RDA_TimeLine_Array[ULN_name, 5]
```

The LOC table element OP_RTG_TRNPN_LOC is a key element and contains the following values:

Value	Meaning
A	Origin
B	Alternate origin
C	ILOC before POE
D	Alternate ILOC before POE
E	POE
F	Alternate POE
G	ILOC before POD
H	Alternative ILOC before POD
J	POD
K	Alternative POD
L	ILOC before destination
M	Alternate ILOC before destination
N	Destination

Also, oplan C date is obtained from table oplan, element op_exn_dt.

Useful properties:

"#RDArqmtList" of app "TL"	List of all rqmts in collection.
"#RDAtypeList" of app "TL"	List of all type codes in collect
"#bufferStart" of page "Main"	The first rqmt in the buffer. This is and integer from 1 to (#numDataRows-#RDAbufferSize).

"#RDAbufferOffset" of pg "Main"	The number of rqmts back the buffer should begin at. The default is 100. For example, if rqmt 5000 was selected, the buffer would begin at rqmt 4900 (5000-\"#RDAbufferOffset\").
"#RDAbufferSize" of page "Main"	The buffer size (Default=1000).
"#numDataRows" of box"ScrollBox" of page "Main"	Number of rqmts in the collection
"#scrollRow" of box "ScrollBox" of page "Main"	The seq # (1 to #numDataRows) of the top rqmt being displayed in timeline.
"#displayRows" of boxWidget "ScrollerFrame" of box "ScrollBox" of page "Main"	The number of rows displayed in timeline. The default is 5.

B.49 RDAToolBox

This application provides a central location to hold common functions that are used frequently within the RDA environment. In addition, it holds global values in properties that are accessed by multiple applications.

B.50 rda_pk_ve1

The rda_pk_ve1 functional unit is part of the verification engine logical unit which performs data checks that are designed to find logical inconsistencies or missing data in requirements. Rda_pk_ve1 specifically contains data check routines common to CIN, PIN and ULN processing including GEOLOC edits. It also contains the OPLAN and Collection driver routines.

B.51 rda_pk_ve2

The rda_pk_ve2 functional unit is part of the verification engine logical unit which performs data checks that are designed to find logical inconsistencies or missing data in requirements. Rda_pk_ve2 specifically contains edit routines unique to CIN processing. There are two driver routines; rda_ve_cin_driver for single CIN processing, and rda_ve_oplan_cin_driver which processes all the CINs for an operations plan.

B.52 rda_pk_ve3

The rda_pk_ve3 functional unit is part of the verification engine logical unit which performs data checks that are designed to find logical inconsistencies or missing data in requirements. Rda_pk_ve3 specifically contains edit routines that are unique to PIN processing as well as the nonunit location edit routines. There are 3 driver routines including rda_ve_pin_driver for single PIN processing, rda_ve_oplan_pin_driver for all PINs for an OPLAN, and rda_ve_nonunit_loc_val which performs nonunit location edits.

B.53 rda_pk_ve4

The rda_pk_ve4 functional unit is part of the verification engine logical unit which performs data checks that are designed to find logical inconsistencies or missing data in requirements. Rda_pk_ve4 specifically contains force (ULN) location edits. There is one driver routine rda_ve_uln_loc_val. The code for the ULN location edits is a copy of the nonunit location edits with a few differences.

B.54 rda_pk_ve5

The rda_pk_ve5 functional unit is part of the verification engine logical unit which performs data checks that are designed to find logical inconsistencies or missing data in requirements. Rda_pk_ve5 specifically contains most of the ULN edit routines. It has two driver routines; rda_ve_uln_driver for single ULN processing, and rda_ve_oplan_uln_driver for multi ULN processing.

B.55 rda_pk_ve6

The rda_pk_ve6 functional unit is part of the verification engine logical unit which performs data checks that are designed to find logical inconsistencies or missing data in requirements. Rda_pk_ve6 specifically contains the edit routines for the OPLAN and its associated records such as oplan_assumption and routines for editing force module information. There are two driver routines; rda_ve_oplan_val for OPLAN edits, and rda_ve_oplan_fm_driver which edits all the force modules for a given OPLAN.

B.56 rda_pk_ve7

The rda_pk_ve7 functional unit is part of the verification engine logical unit which performs data checks that are designed to find logical inconsistencies or missing data in requirements. Rda_pk_ve7 specifically contains the remaining ULN edit routines. It was created to reduce the size of rda_pk_ve5. It has no driver modules since all the drivers are in rda_pk_ve5.

B.57 RenumberOps

This functionality permits a user to renumber or duplicate a set of requirements, and permits the numbering for creating requirements.

B.58 RequirementsUtility

RequirementsUtility checks to see if a particular requirement is contained in a force module. If the requirement is in a force module, RequirementsUtility makes the object passed to it enabled. Otherwise the object is disabled.

B.59 ScriptArchiver

This tool provides the ability to archive the scripts on objects using a hierarchial fashion.

B.60 ScriptTool

This tool provides the ability to browse through scripts on any selected object.

B.62 Select

The Select application allows the user to choose search criteria for constructing a queries relating to requirements. The results of the queries populate what's known as the user's collection. The screen is made up of three main sections. The left side displays the menu traversal of possible choices and in some instances the finite list of values allowed. Each menu choice leaf button relates to an Ad Hoc Query (AHQ) code number that in turn relates to a field and table in the database. As the user picks them, the menu choice along with the specified value are place in the criteria area of the screen. At this point the user may change the operator from the default "=" condition to one of the other allowable ones. The user may also choose sort criteria in the same manner and that is placed in the sort area of the screen. Once all choices are made and Apply is pressed the SQL is generated for the criteria listed. The SQL is executed and the RDA_COLLECTION table is populated with the rows from OPLAN_FORCE_RQMT, OPLAN_NONUNIT_RQMT_CARGO, and OPLAN_NONUNIT_RQMT_PRSL that meet the criteria. Then the sort criteria is applied to the rows in the collection by populating the sort columns in the RDA_COLLECTION table.

B.63 SelectHelper

This tool is provided to aid in the maintenance of the RDASelect work package.

B.64 ShiftTPFDDdates

ShiftTPFDDdates provides the RDA user the ability to make mass date changes (Ready to Load Date ((RLD), Available to Load Date (ALD), Earliest Arrival Date (EAD), Latest Arrival Date (LAD), and Required Delivery Date (RDD)) to the TPFDD. The user selects the dates to be shifted, the number of days to adjust the date(s), and whether the change is to effect the collection, the marked records, or only requirements from a specific force module(s). If "Collection" or "Marked Records" are selected, the following transactions are generated:

a. JJDSDT (Transaction Type A) - To create a temporary FM with a title.

b. JJDSDT (Transaction Type F, G, J) - To attach the requirements in the collection/marked records to the temporary FM.

c. DATEBT - To make the date change for all records with the temporary FM.

d. DLFMDT - To delete the temporary FM.

If the user specified "FMs," a DATEBT transaction is generated for each force module selected. Once the transactions are generated, ShiftTPFDDdates makes a system call to the 'xtp' software which processes the transactions and updates both the local and remote databases. If there are no errors in processing the transactions, xtp returns a code of 0 and the timeline is refreshed to reflect the new dates. If there were processing errors, xtp returns the following codes:

1 = An Unknown Error was Encountered,
2 = Invalid Arguments were Provided, and
3 = An Invalid Transaction was Encountered.

If the return code is 3, ShiftTPFDDdates will display a window which allows the user to see the xtp output files indicating which transactions were in error and why. The output files from xtp are written to the user's home directory and include the following:

a. shiftDates.log - Message log containing messages that are generated for both invalid and valid transactions.

b. shiftDates.tran - Contains the actual transactions generated by ShiftTPFDDdates.

c. shiftDates.rpt - The final report of the transaction results.

d. <userid>_sm_err.lis - Contains errors encountered during processing, not transaction errors.

B.65 SplitShipments

This permits the user to split a record into air (personnel) and sea (cargo) portions. Two requirements are created from the base ULN, one ULN containing the personnel movements (typically an air move) and a second ULN containing the cargo movements (typically a sea move).

B.66 SQLAccess

This is a helper application which contains routines for database access and stored procedure usage through Gain Momentum.

B.67 SQLError

This FU is the standard method for capturing a SQL error that occurs in the database and displaying it to the user. The message is displayed with a standard dialog and the error is recorded in the RDA error log file in the user's home directory.

B.68 SQLOpsTool

This tool permits a user to monitor the number and status of connections between a Gain application and the ORACLE database. Facilities are provided for examining the details of a connection, terminating a connection, and the ability to pass a connection to the SQL Tester application. This final capability is unique in that it allows a developer to open a SQL*PLUS like command line on any given database session. With this capability, the developer is able to look at an in process SQL session before it has been committed.

B.69 SQLTester

This application is a Gain Tool to help the developers monitor and manage their database logins. The tool can be accessed from the ButtonBar Functional Unit. When the tool first comes up, it displays a list of the active SQL logins and certain information about the logins. The developer can select a specific login ID and bring the SQLOpsTool Function Unit, using the selected login ID, to check database changes made that may not have been committed yet. The developer can also look at any pending SQL statements that have not been deallocated, and save them to a file and/or deallocate them. Cursor ID's opened from Gain that have not been closed can also be closed.

B.70 SystemMonitor

This tool provides statistical information about a Gain session to aid in performance tuning.

B.71 TableViewer

This application is a database tool built in Gain, which the developers can use to view any accessible table in the database. It allows the user to specify which columns to display. Simple queries can be built by entering strings in the displayed column fields in "criteria" mode. Executing the query returns the ORACLE ROWIDs of all the matching rows, which can then be viewed one row at a time.

B.72 TargetPIDSelect

This application displays a list of OPLANs for the user to choose as a target to copy requirements or FMs to. This screen works just like the RDASkeleton screen.

B.73 TestApp

This application contains two main pages. The first page records data passed in by a function call "RecordData()," the second page is the RDA version of the GainMomentum message box. The RecordData function is used by various applications when in test mode (CargoEditor and UpdatePIDfromTUCHA) to record data such as SQL statements, etc. This Functional Unit also has the capability of reading in text from a UNIX file, as well as saving the recorded data to a UNIX file.

B.74 TimelineWindows

TimelineWindows is the application which contains most of the pop up windows displayed for Timeline Node and Link Operation. When the RDAUser clicks on the timeline icons pop up windows from application TimelineWindows are displayed. The minitimeline window, and the "RDA: Calendar Day" windows are also contained in TimelineWindows. TimelineWindows allows the RDA user to modify the database and also the Timeline icons through these pop up windows. The RDA user can only change these values if these entries are not locked. TimelineWindows provides error checking of the user's proposed changes and will not allow any invalid data to be sent to the database. If the RDA user enters invalid data he will be told of his error and the data will be reset to the value it was prior to his change. For each change the RDA user applies, the appropriate transactions (Force Requirement Transaction (ULNUBT) for ULNs or Non-Unit Requirement Transaction (NRNUBT) for CINs & PINs) is sent to the send queue.

B.75 TKProfTool

This application will accept an ORACLE generated trace file and run the ORACLE tool TKProf and then display the results of the run to the developer.

B.76 TLUlnList

TLUlnList allows the Timeline Node and Link Operation to be refreshed. It also serves the developer tool "RQMTList" found on the button bar.

B.77 TUCHASelect

This application is used to access the TUCHA reference file (UNIT_TYPE table) to review the data associated with a UTC. Once the desired UTC is found, the application passes the requested information back to the calling function. The queries are built (emulating the GainMomentum data manager query conventions) by calling a function in the RDAToolBox functional unit. The UTCs matching the query criteria are then put into a list box, which can be used to select and view individual records.

B.78 ULNDetail

ULNDetail allows a user to edit an individual ULN. Data to be edited is pulled from OPLAN_FORCE_RQMT and OPLAN_FORCE_RQMT_LOC. Other data displayed comes from UNIT_TYPE and the GSORTS.BIDE table. Edit checks are applied on all fields for data type and, where applicable code values. If the UTC is changed, then cargo and personnel data from the TUCHA is populated in the screen. If the Unit Identification Code (UIC) is changed, then GSORTS data is populated into the Unit Name, Component and Origin fields. If the OPLAN is locked and the ULN is validated or has another pertinent TCC status, then certain cargo and routing data cannot be changed. After pressing the Apply button the edited changes will be applied to the database and a call through RDAToolBox will be made to the single ULNUBT transaction generation script.

B.79 ULNTransactions

This is a helper application for the RDARequirements library. It contains routines that help in the creation, deletion and modification of ULNs, CINs, and PINs.

B.80 UnsplitShipments

This permits the user to rejoin ULN records that were previously split into personnel and cargo components. The C and P records are removed and joined into one ULN.

B.81 UpdatePIDfromTucha

The purpose of this application is to update the data associated with the Force Requirements or Unit Line Numbers (ULNs), in a selected OPLAN, that are using standard data from the TUCHA reference table (UNIT_TYPE). The user accesses this application directly from the main Requirements Development and Analysis (RDA) screen, after selecting the desired OPLAN, and then clicking on "Selected >> Update PID from TUCHA." If the TUCHA table has been updated and changes were made to UTC's being used by a selected OPLAN, this application will identify the differences in the data by displaying three main categories of ULNs affected:

a. ULNs with Deleted UTCs -- The ULNs listed in this category are using UTCs that are no longer found in the TUCHA file, and therefore no Cargo data exists for them.

b. ULNs with Cancelled UTCs -- The ULNs listed in this category are using UTCs that were found in TUCHA, but have a status of "Cancelled."

c. ULNs with Changed UTCs -- The remaining ULNs have UTCs that were found in TUCHA, however, the data associated with the ULN does not match that found in TUCHA, therefore an update is required to synchronize the OPLAN with TUCHA.

The user is given an opportunity to view and/or print the various ULN lists generated. The user also has the option to perform the update at this time. The update can be a partial update, where only the "Changed" ULNs are updated, or a complete update where "Changed" ULNs are updated and the "Cancelled" UTCs are replaced with replacement UTCs (if available) and their associated data updated. Once the complete update is executed the OPLAN's TUCHA date is set to the latest TUCHA date, indicating that the OPLAN has been synchronized with the current TUCHA.

The Update process executes in the background. When done, it creates a pop-up window (if the GAIN session is still active) and sends e-mail to the user.

B.82 VerificationEngine Interface

Application VerificationEngine is the gain interface to access the verification engine stored procedures and tables. VerificationEngine Interface has an option to return the error flag status (none, logical, fatal, or both) and the error message text (i.e. "EAD must be before LAD").

B.83 ViewCheckOuts

This application is only used in the development environment. It allows a developer to list the applications that are checked out.

B.84 ViewGeoloc

This functional unit provides a method for the user to view directly the Standard Geographic Location Reference File. Selection criteria and data is contained on one screen.

B.85 ViewPorts

This functional unit provides a method for the user to view directly the Standard Seaport Reference File. Navigation between screens can be done using pulldown menus and port selection criteria can be entered at the start of the application.

B.86 ViewTucha

This functional unit provides a method for the user to view directly the Standard Type Unit Characteristics Reference File. Navigation between screens supports the display of UTC Replacement, Level III, and Level IV cargo.

B.87 ViewTudet

This functional unit provides a method for the user to view directly the Standard Type Unit Equipment Detail Reference File. Selection criteria and data is contained on one screen.

B.88 XTPTool

This application allows a developer to run a file of transactions through XTP to test the processing of the transaction.